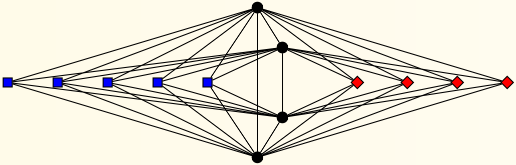


Exact Crossing Number Parameterized by Vertex Cover

Petr Hliněný

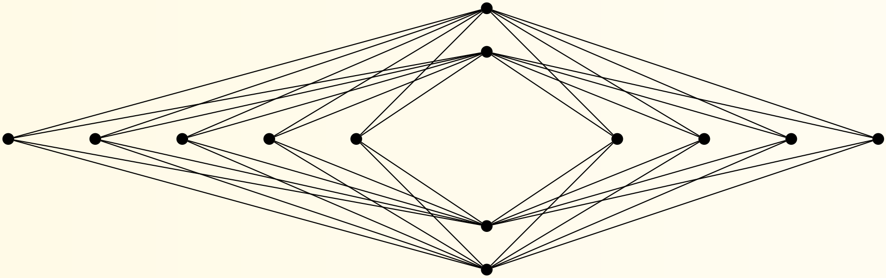
Faculty of Informatics, Masaryk University
Brno, Czech Republic

joint work with

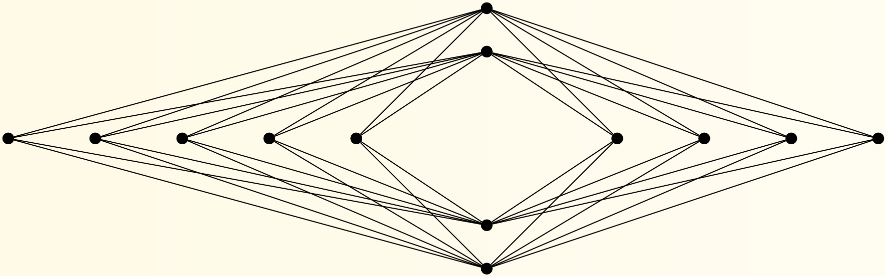
Abhisekh Sankaran

Department of Computer Science and Technology
University of Cambridge, UK

1 Crossing Number Problem

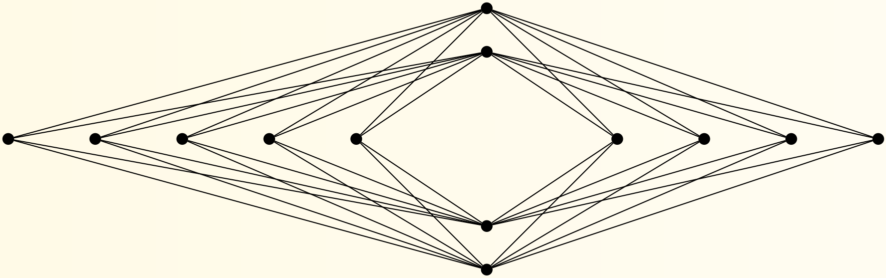


1 Crossing Number Problem



Definition. $CR(m) \equiv$ the problem to draw a graph with $\leq m$ *edge crossings*.

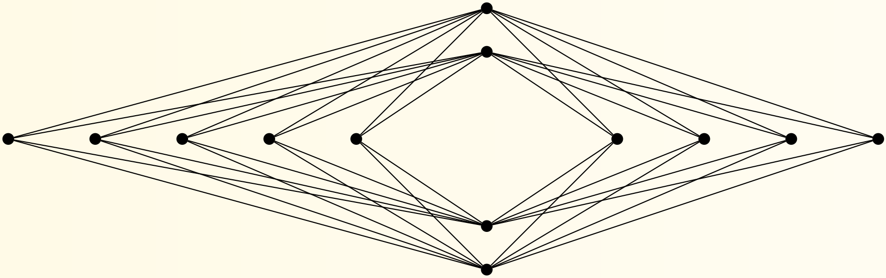
1 Crossing Number Problem



Definition. $CR(m) \equiv$ the problem to draw a graph with $\leq m$ *edge crossings*.

– The vertices of G are distinct points in the plane,
and every edge $e = uv \in E(G)$ is a simple (cont.) curve joining u to v.

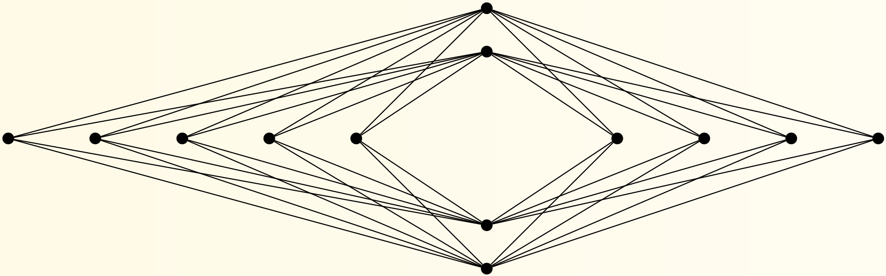
1 Crossing Number Problem



Definition. $CR(m) \equiv$ the problem to draw a graph with $\leq m$ *edge crossings*.

- The vertices of G are distinct points in the plane, and every edge $e = uv \in E(G)$ is a simple (cont.) curve joining u to v.
- No edge passes through a vertex other than its endpoints, and no three edges intersect in a common point.

1 Crossing Number Problem



Definition. $CR(m) \equiv$ the problem to draw a graph with $\leq m$ *edge crossings*.

– The vertices of G are distinct points in the plane, and every edge $e = uv \in E(G)$ is a simple (cont.) curve joining u to v.

– No edge passes through a vertex other than its endpoints, and no three edges intersect in a common point.

- A very hard algorithmic problem, indeed…

Brief complexity status of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]

Brief complexity status of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]

Brief complexity status of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]

Brief complexity status of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]
- And even for *almost-planar* (planar graphs plus one edge)!
  [**Cabello and Mohar**, 2010]

Brief complexity status of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]
- And even for *almost-planar* (planar graphs plus one edge)! [**Cabello and Mohar**, 2010]

Approximations, at least?

- Up to factor $\log^3 |V(G)|$ ($\log^2 \cdot$) for $cr(G) + |V(G)|$ with bounded degs.; [**Even, Guha and Schieber**, 2002]

Brief complexity status of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]
- And even for *almost-planar* (planar graphs plus one edge)! [**Cabello and Mohar**, 2010]

Approximations, at least?

- Up to factor $\log^3 |V(G)|$ $(\log^2 \cdot)$ for *cr*$(G) + |V(G)|$ with bounded degs.; [**Even, Guha and Schieber**, 2002]
- No constant factor approximation for some $c > 1$; [**Cabello**, 2013].

Brief complexity status of $CR(k)$

NP-hardness

- The general case (no surprise); [**Garey and Johnson**, 1983]
- The degree-3 and *minor-monotone* cases; [**PH**, 2004]
- With fixed *rotation scheme*; [**Pelsmajer, Schaeffer, Štefankovič**, 2007]
- And even for *almost-planar* (planar graphs plus one edge)! [**Cabello and Mohar**, 2010]

Approximations, at least?

- Up to factor $\log^3 |V(G)|$ $(\log^2 \cdot)$ for $cr(G) + |V(G)|$ with bounded degs.; [**Even, Guha and Schieber**, 2002]
- No constant factor approximation for some $c > 1$; [**Cabello**, 2013].

Parameterized complexity

- Yes, $CR(k)$ in FPT with parameter k, $\mathcal{O}\big(f(k) \cdot n\big)$ runtime; [**Grohe**, 2001 / **Kawarabayashi and Reed**, 2007]

What about polynomial algorithms?

- Trivially for $CR(c)$ with any constant c (even without the FPT result); just guess the c crossings and test planarity.

What about polynomial algorithms?

- Trivially for $CR(c)$ with any constant c (even without the FPT result); just guess the c crossings and test planarity.
- Even for graphs of tree-width 3, the complexity of $CR(m)$ is unknown!

What about polynomial algorithms?

- Trivially for $CR(c)$ with any constant c (even without the FPT result); just guess the c crossings and test planarity.
- Even for graphs of tree-width 3, the complexity of $CR(m)$ is unknown!
- So, can we come up with any nontrivially rich graph class with unbounded crossing number for which $CR(m)$ is in P (with m on the input)?

What about polynomial algorithms?

- Trivially for $CR(c)$ with any constant c (even without the FPT result); just guess the c crossings and test planarity.
- Even for graphs of tree-width 3, the complexity of $CR(m)$ is unknown!
- So, can we come up with any nontrivially rich graph class with unbounded crossing number for which $CR(m)$ is in P (with m on the input)?

  So far, only one such published result for the *maximal graphs of path-width* 3 by [**Biedl, Chimani, Derka, and Mutzel**, 2017].

What about polynomial algorithms?

- Trivially for $CR(c)$ with any constant c (even without the FPT result); just guess the c crossings and test planarity.
- Even for graphs of tree-width 3, the complexity of $CR(m)$ is unknown!
- So, can we come up with any nontrivially rich graph class with unbounded crossing number for which $CR(m)$ is in P (with m on the input)?

  So far, only one such published result for the *maximal graphs of path-width 3* by [**Biedl, Chimani, Derka, and Mutzel**, 2017].
- **Our contribution:**

  $CR(m)$ is in FPT when parameterized by the vertex cover size.

  (Any m. Warning: only for simple graphs.)

What about polynomial algorithms?

- Trivially for $CR(c)$ with any constant c (even without the FPT result); just guess the c crossings and test planarity.
- Even for graphs of tree-width 3, the complexity of $CR(m)$ is unknown!
- So, can we come up with any nontrivially rich graph class with unbounded crossing number for which $CR(m)$ is in P (with m on the input)?

  So far, only one such published result for the *maximal graphs of path-width* 3 by [**Biedl, Chimani, Derka, and Mutzel**, 2017].
- **Our contribution:**

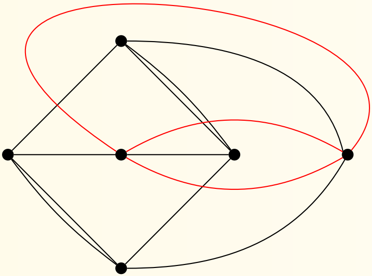
  $\boldsymbol{CR(m)}$ **is in FPT when parameterized by the vertex cover size.**

  (Any m. Warning: only for simple graphs.)

  FPT runtime: $f(k) \cdot n^{\mathcal{O}(1)}$, where $k = |X|$ is the vertex-cover size and f is a computable function (doubly-exponential here).

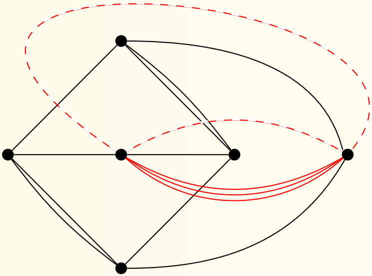
2 Some Basic Ideas

Inspiration: Crossings and parallel edges



2 Some Basic Ideas

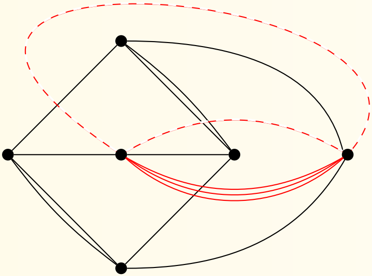
Inspiration: Crossings and parallel edges



Claim.
A bunch of parallel edges can always be *optimally* drawn as one "thick" edge.

2 Some Basic Ideas

Inspiration: Crossings and parallel edges



Claim.

A bunch of parallel edges can always be *optimally* drawn as one "thick" edge.

Proof: Draw whole bunch closely along any of its edges with the least crossings.

Vertex cover and neighbourhood clusters

Vertex cover (VC) $\equiv$ min. number of vertices that hit all edges.

Vertex cover and neighbourhood clusters

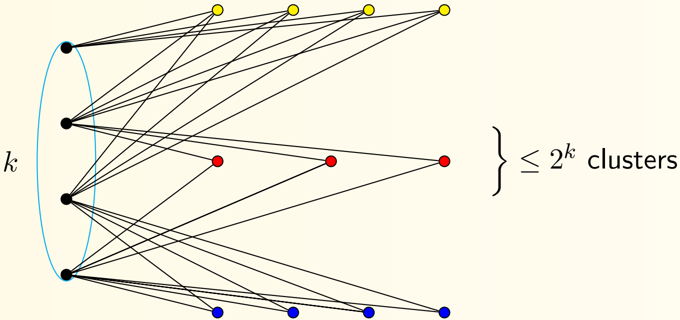
Vertex cover (VC) $\equiv$ min. number of vertices that hit all edges.

(We can compute VC in FPT, even practically. . .)

Vertex cover and neighbourhood clusters

Vertex cover (VC) $\equiv$ min. number of vertices that hit all edges.

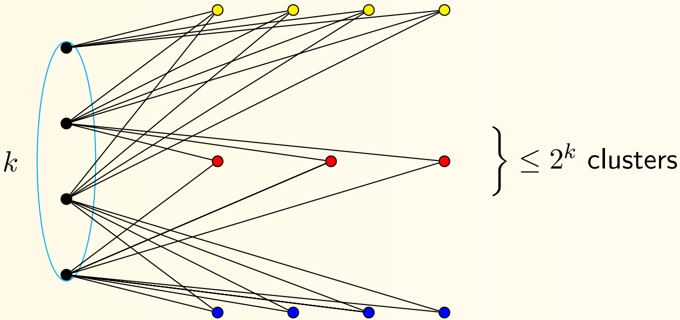
(We can compute VC in FPT, even practically...)



Vertex cover and neighbourhood clusters

Vertex cover (VC) $\equiv$ min. number of vertices that hit all edges.

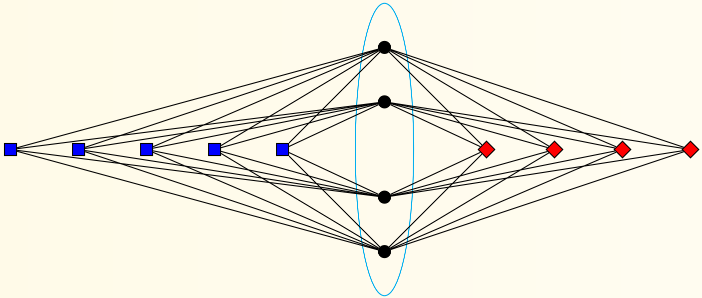
(We can compute VC in FPT, even practically. . .)



- Can we not now just take one neighbourhood cluster and draw it whole closely along its star with the least crossings?

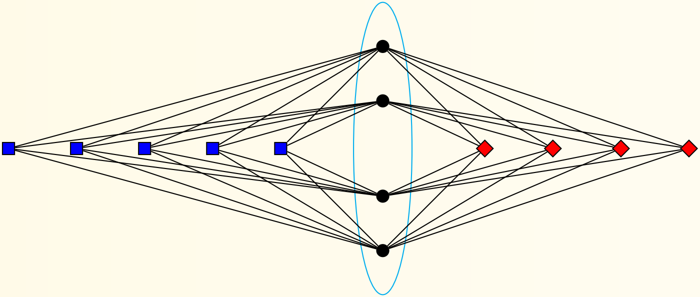
- **NO**, that would be too easy, right?

- **NO**, that would be too easy, right?



- The (unavoidable) fundamental difference between the blue and the red vertices (of $K_{4,9}$ in this case) in an optimal drawing is in the cyclic order of their neighbours.

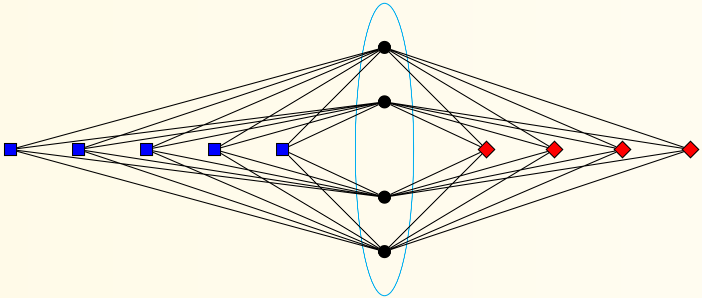
- **NO**, that would be too easy, right?



- The (unavoidable) fundamental difference between the blue and the red vertices (of $K_{4,9}$ in this case) in an optimal drawing is in the cyclic order of their neighbours.

- Surprisingly, this (i.e., neighbours and their cyclic order) is enough!

- **NO**, that would be too easy, right?



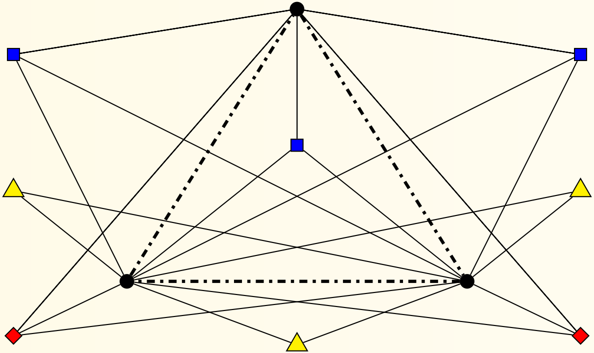
- The (unavoidable) fundamental difference between the blue and the red vertices (of $K_{4,9}$ in this case) in an optimal drawing is in the cyclic order of their neighbours.

- Surprisingly, this (i.e., neighbours and their cyclic order) is enough!

- Rediscovering an idea used for $K_{m,n}$ already by [**Christian, Richter and Salazar**, 2013: Zarankiewicz's Conjecture Is Finite for Each Fixed m].

3 Formal View: Topological Clustering

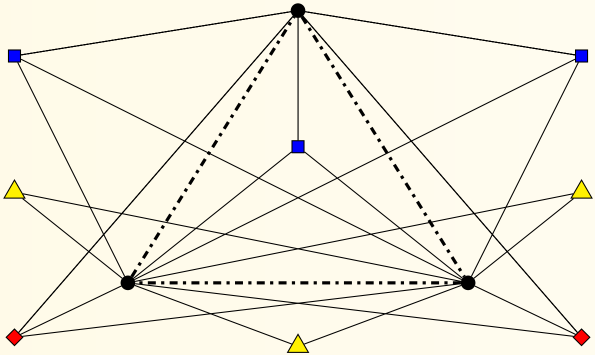
Topological clusters in a drawing



A graph G with a vertex cover X, and its drawing D;

3 Formal View: Topological Clustering

Topological clusters in a drawing

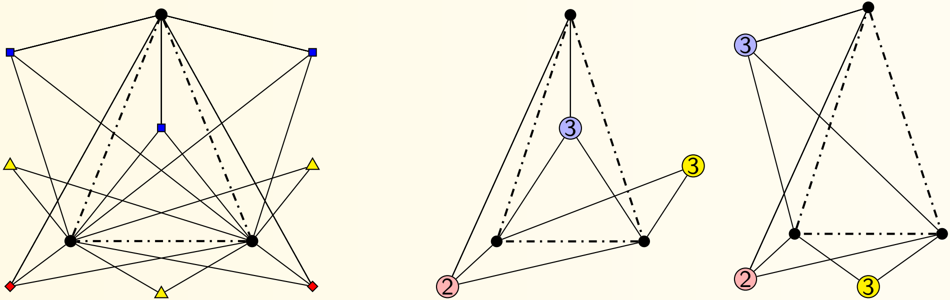


A graph G with a vertex cover X, and its drawing D;

same neighbourhood + same clockwise order in D $\leftrightarrow$ same *topological cluster*

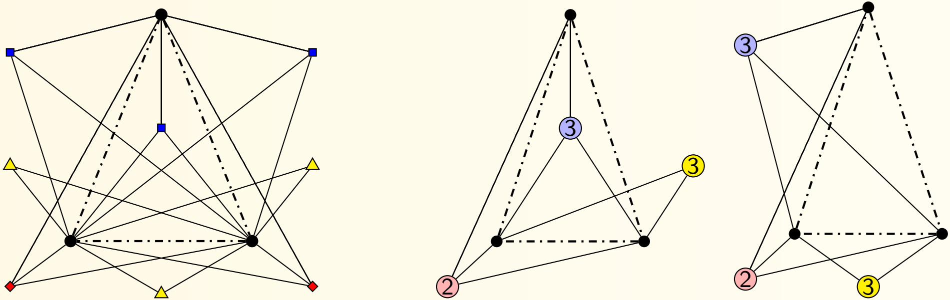
(an equivalence relation on $V(G) \setminus X$).

Topological clustering of a drawing



Topological clustering $\equiv$ an induced subdrawing of D s.t.

Topological clustering of a drawing



Topological clustering $\equiv$ an induced subdrawing of D s.t.

- we pick exactly one representative from each topological cluster of D,
- and remember the size of each cluster as the *weight* of the representative.

The Core Lemma

Consider a drawing D of a graph G, and define

The Core Lemma

Consider a drawing D of a graph G, and define

cluster crossings $\equiv$ those between edges incident with same-cluster vertices,

The Core Lemma

Consider a drawing D of a graph G, and define

cluster crossings $\equiv$ those between edges incident with same-cluster vertices,

non-cluster crossings $\equiv$ all other ones.

The Core Lemma

Consider a drawing D of a graph G, and define

cluster crossings $\equiv$ those between edges incident with same-cluster vertices,

non-cluster crossings $\equiv$ all other ones.

Lemma. For every good drawing D of a graph G with a vertex cover X, there exists its topological clustering D_X such that the number of non-cluster crossings in D is at least $cr(D_X)$ (with weighted crossings!).

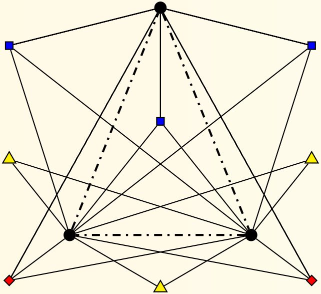
The Core Lemma

Consider a drawing D of a graph G, and define

cluster crossings $\equiv$ those between edges incident with same-cluster vertices,

non-cluster crossings $\equiv$ all other ones.

Lemma. For every good drawing D of a graph G with a vertex cover X, there exists its topological clustering D_X such that the number of non-cluster crossings in D is at least $cr(D_X)$ (with weighted crossings!).



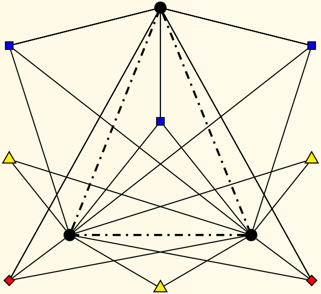
The Core Lemma

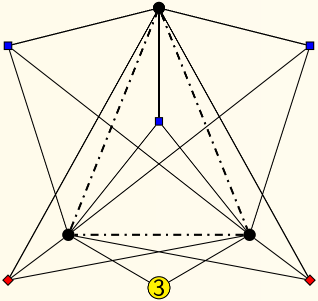
Consider a drawing D of a graph G, and define

cluster crossings $\equiv$ those between edges incident with same-cluster vertices,

non-cluster crossings $\equiv$ all other ones.

Lemma. For every good drawing D of a graph G with a vertex cover X, there exists its topological clustering D_X such that the number of non-cluster crossings in D is at least $cr(D_X)$ (with weighted crossings!).





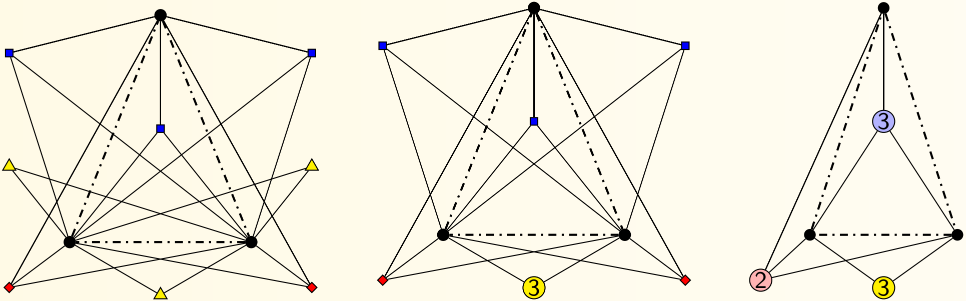
The Core Lemma

Consider a drawing D of a graph G, and define

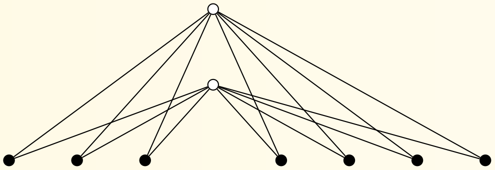
cluster crossings $\equiv$ those between edges incident with same-cluster vertices,

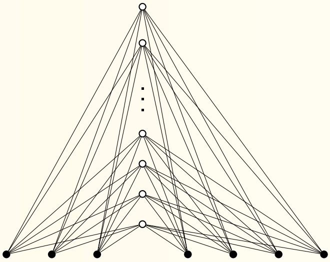
non-cluster crossings $\equiv$ all other ones.

Lemma. For every good drawing D of a graph G with a vertex cover X, there exists its topological clustering D_X such that the number of non-cluster crossings in D is at least $cr(D_X)$ (with weighted crossings!).



Counting Cluster Crossings

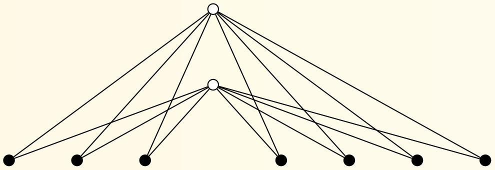


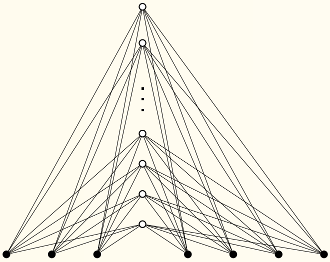


Lemma. [**Christian, Richter and Salazar**, 2013]
Any drawing of $K_{2,m}$ that has the same clockwise cyclic order in the part with 2 vertices has at least

$$\left\lfloor \frac{m}{2} \right\rfloor \cdot \left\lfloor \frac{m-1}{2} \right\rfloor \text{ crossings.}$$

Counting Cluster Crossings





Lemma. [**Christian, Richter and Salazar**, 2013]
Any drawing of $K_{2,m}$ that has the same clockwise cyclic order in the part with 2 vertices has at least

$$\left\lfloor \frac{m}{2} \right\rfloor \cdot \left\lfloor \frac{m-1}{2} \right\rfloor \text{ crossings.}$$

Corollary. Any topological cluster of size (weight) c and with m neighbours in X has at least

$$\binom{c}{2} \cdot \left\lfloor \frac{m}{2} \right\rfloor \cdot \left\lfloor \frac{m-1}{2} \right\rfloor \text{ (cluster) crossings.}$$

4 Algorithmic Side: Brute Force and IQP

Step I: Abstract topological clusterings

I.e., topological clusterings of some drawing of G, stripped of their weights.

4 Algorithmic Side: Brute Force and IQP

Step I: Abstract topological clusterings

I.e., topological clusterings of some drawing of G, stripped of their weights.

Lemma. There are only $2^{k^{\mathcal{O}(k)}}$ possible non-equivalent planarizations of the abstract topological clusterings of G.

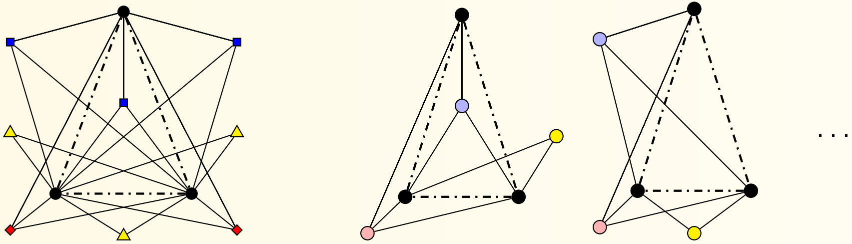
4 Algorithmic Side: Brute Force and IQP

Step I: Abstract topological clusterings

I.e., topological clusterings of some drawing of G, stripped of their weights.

Lemma. There are only $2^{k^{\mathcal{O}(k)}}$ possible non-equivalent planarizations of the abstract topological clusterings of G.

$\rightarrow$ We can guess the right one by brute force in FPT!



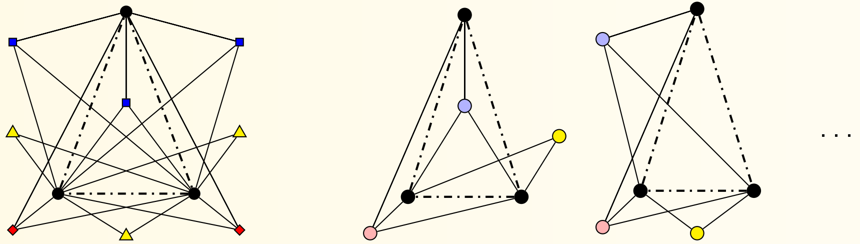
4 Algorithmic Side: Brute Force and IQP

Step I: Abstract topological clusterings

I.e., topological clusterings of some drawing of G, stripped of their weights.

Lemma. There are only $2^{k^{\mathcal{O}(k)}}$ possible non-equivalent planarizations of the abstract topological clusterings of G.

$\rightarrow$ We can guess the right one by brute force in FPT!



$\rightarrow$ But, what about the cluster weights?

Step II: Integer Quadratic Programming

IQP: to find an optimal solution $\boldsymbol{z}^{\circ}$ to the following optimization problem

$$\begin{array}{rrl} \text{Minimize} & \boldsymbol{z}^T\boldsymbol{Q}\boldsymbol{z} & +\,\boldsymbol{p}^T\boldsymbol{z} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{z} & \le\ \boldsymbol{b} \\ & \boldsymbol{C}\boldsymbol{z} & =\ \boldsymbol{d} \\ & \boldsymbol{z} & \in\ \mathbb{Z}^k \end{array}$$

Step II: Integer Quadratic Programming

IQP: to find an optimal solution $\boldsymbol{z}^\circ$ to the following optimization problem

$$\begin{array}{rcl} \text{Minimize} & \boldsymbol{z}^T\boldsymbol{Q}\boldsymbol{z} & +\,\boldsymbol{p}^T\boldsymbol{z} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{z} & \leq\ \boldsymbol{b} \\ & \boldsymbol{C}\boldsymbol{z} & =\ \boldsymbol{d} \\ & \boldsymbol{z} & \in\ \mathbb{Z}^k \end{array}$$

Theorem. [**Lokshtanov**, 2015]
This IQP can be solved in time $f(k,\lambda)\cdot L^{\mathcal{O}(1)}$ where

- $L =$ the length of the combined bit representation of the IQP,
- $\lambda =$ max entry in the matrices $\boldsymbol{A}$, $\boldsymbol{C}$, $\boldsymbol{Q}$ and $\boldsymbol{p}$,
- $k =$ the number of integer variables.

IQP formulation for our case

Suppose an abstract clustering C. What do we have to care about now?

IQP formulation for our case

Suppose an abstract clustering C. What do we have to care about now?

– Every neighbourh. cluster size $g(i)$ $\rightarrow$ partition to weights of its abstract topological clusters (integer vector $\boldsymbol{z}$, with sections for each cluster).

IQP formulation for our case

Suppose an abstract clustering C. What do we have to care about now?

- Every neighbourh. cluster size $g(i) \rightarrow$ partition to weights of its abstract topological clusters (integer vector $\boldsymbol{z}$, with sections for each cluster).
- The weight $z_{(a,b)}$ of each topological cluster contributes to (cluster) crossings by an explicit quadratic formula above.

IQP formulation for our case

Suppose an abstract clustering C. What do we have to care about now?

- Every neighbourh. cluster size $g(i) \rightarrow$ partition to weights of its abstract topological clusters (integer vector $\boldsymbol{z}$, with sections for each cluster).
- The weight $z_{(a,b)}$ of each topological cluster contributes to (cluster) crossings by an explicit quadratic formula above.
- Every actual crossing in C, by the weight(s), contributes an easy quadratic (or linear if one edge in X) term to the total (non-cluster) crossings.

IQP formulation for our case

Suppose an abstract clustering C. What do we have to care about now?

- Every neighbourh. cluster size $g(i)$ $\rightarrow$ partition to weights of its abstract topological clusters (integer vector $\boldsymbol{z}$, with sections for each cluster).
- The weight $z_{(a,b)}$ of each topological cluster contributes to (cluster) crossings by an explicit quadratic formula above.
- Every actual crossing in C, by the weight(s), contributes an easy quadratic (or linear if one edge in X) term to the total (non-cluster) crossings.
- Altogether...

$$
\begin{array}{lcll}
\text{Minimize} & f(\boldsymbol{z}) & = \frac{1}{2}\,\boldsymbol{z}^T \boldsymbol{Q} \boldsymbol{z} + \boldsymbol{p}^T \boldsymbol{z} + c_0 & \\
\text{over all} & \boldsymbol{z} & = \left(z_{(1,1)}, \dots, z_{(1,g(1))}, \dots, z_{(l,1)}, \dots, z_{(l,g(l))}\right) & \\
\text{subject to} & \sum_{j=1}^{g(i)} z_{(i,j)} & = g(i) & \text{for } i \in \{1, \dots, l\} \\
 & z_{(i,j)} & \geq 0 & \text{for } (i,j) \in I = \{(1,1), \dots, (l, g(l))\} \\
 & \boldsymbol{z} & \in \mathbb{Z}^{|I|} &
\end{array}
$$

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.
- Multigraphs? Those bring various deep problems, e.g. . .

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.
- Multigraphs? Those bring various deep problems, e.g. . .
  - getting an unbounded number of neighbourhood clusters, and

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.
- Multigraphs? Those bring various deep problems, e.g...
  - getting an unbounded number of neighbourhood clusters, and
  - getting too high entries in the matrix of our IQP.

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.

- Multigraphs? Those bring various deep problems, e.g. . .
  - getting an unbounded number of neighbourhood clusters, and
  - getting too high entries in the matrix of our IQP.

We actually believe the non-simple variant to be W-hard.

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.
- Multigraphs? Those bring various deep problems, e.g. . .
  - getting an unbounded number of neighbourhood clusters, and
  - getting too high entries in the matrix of our IQP.

  We actually believe the non-simple variant to be W-hard.
- Adding “more layers” to our clustering?

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.
- Multigraphs? Those bring various deep problems, e.g. . .
  - getting an unbounded number of neighbourhood clusters, and
  - getting too high entries in the matrix of our IQP.

  We actually believe the non-simple variant to be W-hard.
- Adding "more layers" to our clustering?
  This would look like parameterization by the *tree-depth*, however. . .

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.

- Multigraphs? Those bring various deep problems, e.g...

  - getting an unbounded number of neighbourhood clusters, and
  - getting too high entries in the matrix of our IQP.

  We actually believe the non-simple variant to be W-hard.

- Adding "more layers" to our clustering?
  This would look like parameterization by the *tree-depth*, however...

  - we tried hard (also with other collaborators), but it looks hopeless,

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.
- Multigraphs? Those bring various deep problems, e.g. . .
  - getting an unbounded number of neighbourhood clusters, and
  - getting too high entries in the matrix of our IQP.

  We actually believe the non-simple variant to be W-hard.
- Adding "more layers" to our clustering?
  This would look like parameterization by the *tree-depth*, however. . .
  - we tried hard (also with other collaborators), but it looks hopeless,
  - actually, one would have to do this first for Optimal linear arrangement, and even that seems terribly hard.

5 Conclusions and Questions

- We can compute the exact crossing number parameterized by the vertex cover, but only for simple graphs.
- Multigraphs? Those bring various deep problems, e.g. . .
  - getting an unbounded number of neighbourhood clusters, and
  - getting too high entries in the matrix of our IQP.

  We actually believe the non-simple variant to be W-hard.
- Adding "more layers" to our clustering?
  This would look like parameterization by the *tree-depth*, however. . .
  - we tried hard (also with other collaborators), but it looks hopeless,
  - actually, one would have to do this first for Optimal linear arrangement, and even that seems terribly hard.

Thank you for your attention.